

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



BUILD A SCROLLABLE LIST

Oleh:

Ririn Citra Lestari

NIM. 2310817120012

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
APRIL 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
MODUL 3

Laporan Praktikum Pemrograman Mobile Modul 3: Build a Scrollable List ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Ririn Citra Lestari
NIM : 2310817120012

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Raka Azwar
NIM. 2210817210012

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 201903 01 011

DAFTAR ISI

| | |
|------------------------|----|
| LEMBAR PENGESAHAN..... | 2 |
| DAFTAR ISI..... | 3 |
| DAFTAR GAMBAR..... | 4 |
| DAFTAR TABEL..... | 5 |
| SOAL 1..... | 6 |
| A. SOURCE CODE..... | 7 |
| B. OUTPUT PROGRAM..... | 18 |
| C. PEMBAHASAN | 22 |
| D. TAUTAN GIT..... | 23 |
| SOAL 2..... | 24 |

DAFTAR GAMBAR

| | |
|----------------------------------|----|
| Gambar 1. Contoh UI List..... | 6 |
| Gambar 2. Contoh UI Detail | 7 |
| Gambar 3. Soal 1 (Compose) | 18 |
| Gambar 4. Soal 1 (Compose) | 19 |
| Gambar 5. Soal 1 (Compose) | 19 |
| Gambar 6. Soal 1 (XML) | 20 |
| Gambar 7. Soal 1 (XML) | 20 |
| Gambar 8. Soal 1 (XML) | 21 |
| Gambar 9. Soal 1 (XML) | 21 |

DAFTAR TABEL

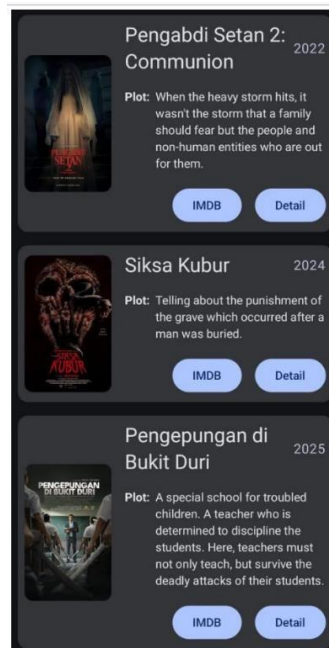
| | |
|--|----|
| Tabel 1. Source Code Jawaban Soal 1 | 7 |
| Tabel 2. Source Code Jawaban Soal 1 | 7 |
| Tabel 3. Source Code Jawaban Soal 1 | 8 |
| Tabel 4. Source Code Jawaban Soal 1 | 9 |
| Tabel 5. Source Code Jawaban Soal 1 | 9 |
| Tabel 6. Source Code Jawaban Soal 1 | 11 |
| Tabel 7. Source Code Jawaban Soal 1 | 12 |
| Tabel 8. Source Code Jawaban Soal 1 | 12 |
| Tabel 9. Source Code Jawaban Soal 1 | 13 |
| Tabel 10. Source Code Jawaban Soal 1 | 15 |
| Tabel 11. Source Code Jawaban Soal 1 | 16 |
| Tabel 12. Source Code Jawaban Soal 1 | 16 |
| Tabel 13. Source Code Jawaban Soal 1 | 17 |

SOAL 1

Soal Praktikum:

Buatlah sebuah aplikasi Android menggunakan XML dan Jetpack Compose yang dapat menampilkan list dengan ketentuan berikut:

1. List menggunakan fungsi RecyclerView (XML) dan LazyColumn (Compose)
2. List paling sedikit menampilkan 5 item. Tema item yang ingin ditampilkan bebas
3. Item pada list menampilkan teks dan gambar sesuai dengan contoh di bawah
4. Terdapat 2 button dalam list, dengan fungsi berikut:
 - a. Button pertama menggunakan intent eksplisit untuk membuka aplikasi atau browser lain
 - b. Button kedua menggunakan Navigation component untuk membuka laman detail item
5. Sudut item pada list dan gambar di dalam list melengkung atau rounded corner menggunakan Radius
6. Saat orientasi perangkat berubah/dirotasi, baik ke portrait maupun landscape, aplikasi responsif dan dapat menunjukkan list dengan baik. Data di dalam list tidak boleh hilang
7. Aplikasi menggunakan arsitektur single activity (satu activity memiliki beberapa fragment)
8. Aplikasi berbasis XML harus menggunakan ViewBinding



Gambar 1. Contoh UI List

Desain UI laman detail bebas, tetapi diusahakan untuk mengikuti kaidah desain Material Design dan data item ditampilkan penuh di laman detail seperti contoh berikut:



Gambar 2. Contoh UI Detail

A. SOURCE CODE

Compose:

MainActivity.kt

Tabel 1. Source Code Jawaban Soal 1

```
package com.example.listfilm

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.navigation.compose.rememberNavController
import com.example.listfilm.ui.screen.AppNavGraph

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            val navController = rememberNavController()
            AppNavGraph(navController)
        }
    }
}
```

FilmData.kt

Tabel 2. Source Code Jawaban Soal 1

```
package com.example.listfilm.data
import com.example.listfilm.R

val filmList = listOf(
    Film(
        title = "Happiness",
        description = "Drama Korea Happiness adalah drama Korea yang
mengusung tema tentang perjuangan manusia menghadapi penyakit menular
di dunia yang terjebak dalam perbedaan kelas sosial.",
        imageResId = R.drawable.happiness,
```

```

        url = "https://www.bilibili.tv/id/video/4792916821151744",
        genre = "Action, Thriller, Drama, Sci-Fi",
        year = 2021
    ),
    Film(
        title = "All Of Us Are Dead",
        description = "Sebuah SMA menjadi titik nol merebaknya wabah virus zombi. Para murid yang terperangkap pun harus berjuang untuk kabur jika tak mau terinfeksi dan berubah menjadi buas.",
        imageResId = R.drawable.aouad,
        url = "https://www.bilibili.tv/id/video/2049830922",
        genre = "Action, Thriller, Horror, Sci-Fi",
        year = 2022
    ),
    Film(
        title = "The Trauma Code",
        description = "Seorang ahli bedah jenius yang tergabung dengan tim trauma parah memimpin sebuah tim untuk berjuang di sebuah rumah sakit universitas. ",
        imageResId = R.drawable.traumacenter,
        url = "https://www.bilibili.tv/id/video/4794455678652416",
        genre = "Action, Drama, Medical, Fantasy",
        year = 2025
    ),
    Film(
        title = "Newtopia",
        description = "sebuah drama yang mengisahkan perjalanan seorang tentara bernama Jae Yoon. Tanpa diduga, ia harus menghadapi wabah zombie bersama mantan kekasihnya, Kang Yeong Ju.",
        imageResId = R.drawable.newtopia,
        url = "https://www.bilibili.tv/id/video/4794465462128640",
        genre = "Action, Thriller, Romance, Fantasy",
        year = 2025
    ),
    Film(
        title = "Snow Drop",
        description = "Drama Snowdrop ini berlatar Korea Selatan pada tahun 1987, menceritakan tentang Im Soo Ho (Jung Hae In) seorang mahasiswa di sebuah universitas bergengsi yang tiba-tiba masuk ke asrama wanita dengan berlumuran darah. Disana ia bertemu dengan seorang mahasiswi Eun Young Ro (Kim Ji Soo) yang ceria dan menyenangkan. Young Ro lantas menyembunyikan Soo Ho dan merawat lukanya bahkan disaat pengawasan ketat.",
        imageResId = R.drawable.snowdrop,
        url = "https://www.bilibili.tv/id/video/2009361160",
        genre = "Romance, Drama, Melodrama, Political",
        year = 2021
    )
)

```


Film.kt

Tabel 3. Source Code Jawaban Soal 1

```
package com.example.listfilm.data

data class Film(
    val title: String,
    val year: Int,
    val genre: String,
```

```

        val description: String,
        val imageResId: Int,
        val url: String
    )

```

FilmDetailScreen.kt

Tabel 4. Source Code Jawaban Soal 1

```

package com.example.listfilm.ui.screen

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import com.example.listfilm.data.Film

@Composable
fun FilmDetailScreen(film: Film) {
    Column(modifier = Modifier.padding(16.dp)) {
        Image(
            painter = painterResource(id = film.imageResId),
            contentDescription = null,
            modifier = Modifier
                .fillMaxWidth()
                .height(200.dp)
                .clip(RoundedCornerShape(12.dp)),
            contentScale = ContentScale.Crop
        )
        Spacer(modifier = Modifier.height(16.dp))
        Text(text = film.title, style =
MaterialTheme.typography.headlineSmall)
        Text(text = "${film.genre} • ${film.year}")
        Spacer(modifier = Modifier.height(8.dp))
        Text(text = film.description)
    }
}

```

FilmListScreen.kt

Tabel 5. Source Code Jawaban Soal 1

```

package com.example.listfilm.ui.screen

import android.content.Intent
import android.net.Uri
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*

```

```

import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.navigation.NavHostController
import androidx.navigation.compose.rememberNavController
import androidx.core.net.toUri
import com.example.listfilm.data.filmList

@Composable
fun FilmListScreen(navController: NavHostController) {
    val context = LocalContext.current
    val films = filmList

    LazyColumn(contentPadding = PaddingValues(16.dp)) {
        items(films) { film ->
            Card(
                shape = RoundedCornerShape(16.dp),
                modifier = Modifier
                    .padding(bottom = 16.dp)
                    .fillMaxWidth()
            ) {
                Column(Modifier.padding(16.dp)) {
                    Image(
                        painterResource(id =
film.imageResId),
                        contentDescription = null,
                        modifier = Modifier
                            .fillMaxWidth()
                            .height(200.dp)
                            .clip(RoundedCornerShape(12.dp)),
                        contentScale = ContentScale.Crop
                    )
                    Spacer(Modifier.height(8.dp))
                    Row(
                        horizontalArrangement =
Arrangement.SpaceBetween,
                        modifier = Modifier.fillMaxWidth()
                    ) {
                        Text(text = film.title, fontWeight =
FontWeight.Bold)

                        Text(text = "${film.year}")
                    }
                }
            }
        }
    }

```

```

        Row(
            horizontalArrangement
Arrangement.SpaceBetween,
            modifier = Modifier.fillMaxWidth()
        ) {
            Text(text = film.genre)
            Text(text = "Korean")
        }
        Spacer(Modifier.height(8.dp))
        Row(
            modifier = Modifier.fillMaxWidth(),
            horizontalArrangement
Arrangement.SpaceBetween
        ) {
            Button(onClick = {
                val intent = Intent(Intent.ACTION_VIEW,
film.url.toUri())
                context.startActivity(intent)
            }) {
                Text("Open Site")
            }
            Button(onClick = {
navController.navigate("detail/${Uri.encode(film.title)}")
            }) {
                Text("Detail")
            }
        }
    }
}

@Preview(showBackground = true)
@Composable
fun FilmListScreenPreview() {
    val navController = rememberNavController()
    FilmListScreen(navController = navController)
}

```

NavGraph.kt

Tabel 6. Source Code Jawaban Soal 1

```

package com.example.listfilm.ui.screen

import android.net.Uri
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.navigation.NavHostController
import androidx.navigation.compose.NavHost

```

```

import androidx.navigation.compose.composable
import com.example.listfilm.data.filmList

@Composable
fun AppNavGraph(navController: NavHostController) {
    val films = filmList // langsung properti filmList
    NavHost(navController = navController, startDestination = "list")
    {
        composable("list") {
            FilmListScreen(navController)
        }
        composable("detail/{title}") { backStackEntry ->
            val rawTitle = backStackEntry.arguments?.getString("title")
            val decodedTitle = rawTitle?.let { Uri.decode(it) }
            val film = films.find { it.title == decodedTitle }
            if (film != null) {
                FilmDetailScreen(film = film)
            } else {
                Text("Film not found", modifier =
                Modifier.fillMaxSize())
            }
        }
    }
}

```

XML:

Activity_main.xml

Tabel 7. Source Code Jawaban Soal 1

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:padding="16dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>

```

activity_detail.xml

Tabel 8. Source Code Jawaban Soal 1

□

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="centerCrop" />

        <TextView
            android:id="@+id/titleText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:textStyle="bold"
            android:layout_marginTop="12dp" />

        <TextView
            android:id="@+id/detailText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp" />
    </LinearLayout>
</ScrollView>

```

item_film.xml

Tabel 9. Source Code Jawaban Soal 1

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:layout_marginTop="8dp"
    android:elevation="4dp"
    android:padding="12dp"
    android:radius="12dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="centerCrop" />

        <TextView
            android:id="@+id/titleText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="16sp"
            android:textStyle="bold"
            android:paddingTop="8dp"/>

        <TextView
            android:id="@+id/yearGenreText"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#888888"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_marginTop="8dp"
            android:gravity="end">

            <Button
                android:id="@+id/detailButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Detail" />

            <Button
                android:id="@+id/siteButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Open Site"
                android:layout_marginStart="8dp"/>

        </LinearLayout>
    </LinearLayout>
</androidx.cardview.widget.CardView>

```

DetailActivity.kt

Tabel 10. Source Code Jawaban Soal 1

```
package com.example.listfilm

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.listfilm.data.filmList
import com.example.listfilm.databinding.ActivityDetailBinding

class DetailActivity : AppCompatActivity() {
    private lateinit var binding: ActivityDetailBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityDetailBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val title = intent.getStringExtra("title")
        val film = filmList.find { it.title == title }

        film?.let {
            binding.imageView.setImageResource(it.imageResId)
            binding.titleText.text = it.title
            binding.detailText.text = "${it.genre} •
            ${it.year}\n\n${it.description}"
        } ?: run {
            binding.titleText.text = "Film not found"
        }
    }
}
```

11

Film.kt

Tabel 11. Source Code Jawaban Soal 1

```
package com.example.listfilm.data

data class Film(
    val title: String,
    val genre: String,
    val year: Int,
    val description: String,
    val imageResId: Int,
    val url: String
)
```

FilmAdapter.kt

Tabel 12. Source Code Jawaban Soal 1

```
package com.example.listfilm

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example.listfilm.data.Film
import com.example.listfilm.databinding.ItemFilmBinding

class FilmAdapter(
    private val filmList: List<Film>,
    private val onDetailClick: (Film) -> Unit,
    private val onOpenSiteClick: (Film) -> Unit
) : RecyclerView.Adapter<FilmAdapter.FilmViewHolder>() {

    inner class FilmViewHolder(private val binding: ItemFilmBinding)
    :
        RecyclerView.ViewHolder(binding.root) {
        fun bind(film: Film) {
            binding.imageView.setImageResource(film.imageResId)
            binding.titleText.text = film.title
            binding.yearGenreText.text = "${film.year} •
            ${film.genre}"
            binding.detailButton.setOnClickListener {
                onDetailClick(film)
            }
            binding.siteButton.setOnClickListener {
                onOpenSiteClick(film)
            }
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType:
    Int): FilmViewHolder {
        val binding =
```

```
ItemFilmBinding.inflate(LayoutInflater.from(parent.context), parent,
false)
    return FilmViewHolder(binding)
}

override fun onBindViewHolder(holder: FilmViewHolder, position:
Int) {
    holder.bind(filmList[position])
}

override fun getItemCount(): Int = filmList.size
}
```

MainActivity.kt

Tabel 13. Source Code Jawaban Soal 1

```
package com.example.listfilm

import android.content.Intent
import android.net.Uri
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.listfilm.data.filmList
import com.example.listfilm.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

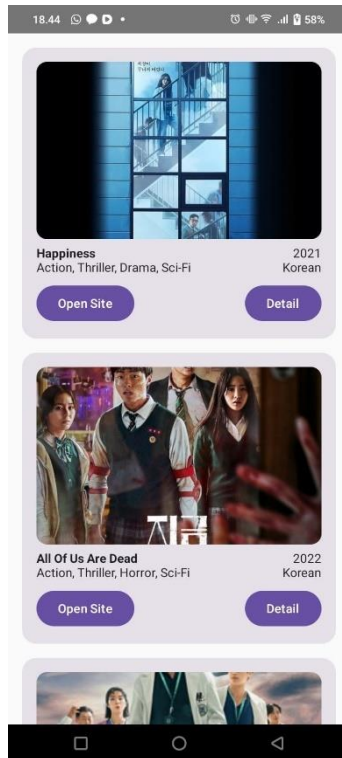
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        val adapter = FilmAdapter(
            filmList,
            onDetailClick = { film ->
                val intent = Intent(this,
DetailActivity::class.java)
                intent.putExtra("title", film.title)
                startActivity(intent)
            },
            onOpenSiteClick = { film ->
                val intent = Intent(Intent.ACTION_VIEW,
Uri.parse(film.url))
                startActivity(intent)
            }
        )

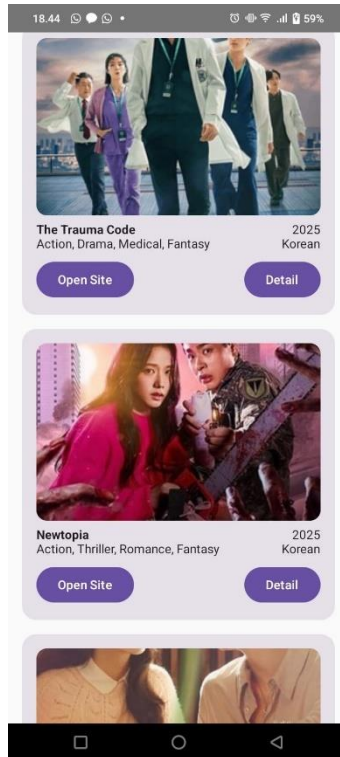
        binding.recyclerView.layoutManager =
LinearLayoutManager(this)
        binding.recyclerView.adapter = adapter
    }
}
```

□

B. OUTPUT PROGRAM



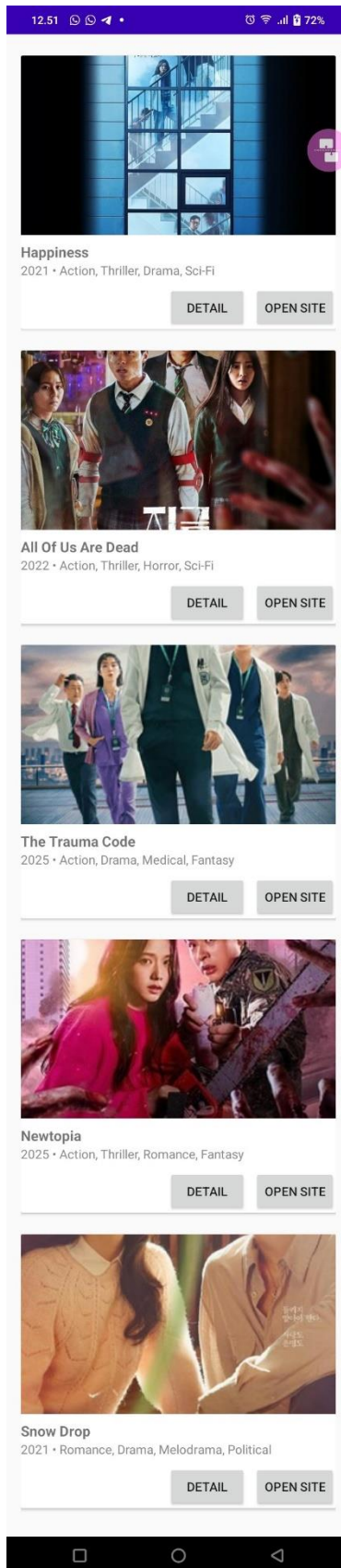
Gambar 3. Soal 1 (Compose)



Gambar 4. Soal 1 (Compose)



Gambar 5. Soal 1 (Compose)



Gambar 6. Soal 1(XML)



C. PEMBAHASAN

Kode di atas merupakan implementasi aplikasi daftar film yang dibangun dengan dua pendekatan antarmuka pengguna di Android: **Jetpack Compose** dan **XML tradisional**. Aplikasi ini memungkinkan pengguna untuk melihat daftar film, melihat detail setiap film, dan membuka situs eksternal terkait film tersebut. Data film disimpan dalam `FilmData.kt` dalam bentuk list `filmList`, yang masing-masing item-nya adalah objek dari kelas `Film`, berisi informasi seperti judul, deskripsi, genre, tahun rilis, URL, dan ID gambar.

Untuk pendekatan **Jetpack Compose**, tampilan utama ditangani oleh `FilmListScreen.kt`, yang menggunakan `LazyColumn` untuk menampilkan daftar film secara efisien. Setiap item ditampilkan dalam komponen `Card`, yang berisi gambar, judul, dan tahun rilis film. Ketika item dipilih, pengguna dapat dinavigasi ke halaman detail melalui `NavHostController`. Tampilan detail film dikelola oleh `FilmDetailScreen.kt`, yang menampilkan gambar besar, judul, genre dan tahun, serta deskripsi film menggunakan elemen-elemen `Compose` seperti `Image`, `Text`, dan `Spacer`. Navigasi antar layar diatur oleh `NavGraph.kt`, meskipun kode lengkapnya tidak ditampilkan.

Sementara itu, dalam pendekatan **XML konvensional**, file seperti `activity_main.xml`, `activity_detail.xml`, dan `item_film.xml` digunakan untuk mendefinisikan tampilan secara statis. Komponen seperti `RecyclerView` digunakan untuk menampilkan daftar film, dan `FilmAdapter.kt` bertanggung jawab atas pengisian data ke dalam item tampilan. `FilmAdapter` mengatur `ViewHolder`, dan pada fungsi `onBindViewHolder`, gambar, judul, dan info film ditampilkan, serta dua tombol diatur: satu untuk membuka link ke situs film menggunakan `Intent.ACTION_VIEW`, dan satu lagi untuk membuka `DetailActivity` yang menampilkan detail film.

Dengan dua pendekatan ini, pengguna dapat menelusuri daftar film, membuka detail, serta menavigasi ke tautan eksternal untuk menonton atau mengetahui lebih lanjut. `Jetpack Compose` menawarkan pendekatan deklaratif modern, sementara `XML` dan `RecyclerView` masih digunakan untuk kompatibilitas dan kontrol langsung terhadap UI. Kombinasi data model (`Film`), pengelolaan UI, dan navigasi membentuk aplikasi yang utuh dan interaktif.

D. TAUTAN GIT

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/rc114/PrakMobile.git>

SOAL 2

Mengapa RecyclerView masih digunakan, padahal RecyclerView memiliki kode yang panjang dan bersifat boiler-plate, dibandingkan LazyColumn dengan kode yang lebih singkat?

Jawabannya :

Menurut saya, alasan kenapa RecyclerView masih digunakan sampai sekarang, meskipun kodenya terkesan panjang dan boiler-plate dibandingkan dengan LazyColumn di Jetpack Compose, ada beberapa poin penting.

Pertama, RecyclerView masih sangat kompatibel dengan View system lama di Android. Banyak aplikasi yang masih menggunakan XML dan belum sepenuhnya migrasi ke Compose, jadi RecyclerView tetap jadi pilihan utama karena bisa langsung diintegrasikan tanpa harus mengubah struktur besar-besaran.

Kedua, RecyclerView menawarkan fleksibilitas dan kontrol yang sangat tinggi. Saya bisa menggunakan LayoutManager custom, animasi transisi item lewat ItemAnimator, atau bahkan pakai SnapHelper. Ini penting banget kalau butuh UI yang kompleks.

Selain itu, karena RecyclerView sudah ada sejak lama, dokumentasinya lengkap, tutorialnya banyak, dan komunitasnya luas. Kalau ada masalah, saya bisa dengan mudah cari solusi di Stack Overflow atau artikel lainnya.

Jujur, meskipun LazyColumn di Compose jauh lebih ringkas dan efisien untuk use case sederhana, tapi Compose sendiri masih terus berkembang dan belum bisa menggantikan semua fitur lanjutan yang dimiliki RecyclerView. Terutama kalau butuh nested scroll yang kompleks atau integrasi dengan komponen View yang lebih lama.

Jadi menurut saya, RecyclerView masih sangat relevan, khususnya untuk proyek-proyek besar atau aplikasi yang belum full Compose. Tapi kalau saya bikin aplikasi baru dan pengen UI modern yang lebih cepat dikembangkan, tentu saya bakal pilih LazyColumn.